

Dossier de Conception Système : IKEBay

Yann Ducruy

20 décembre 2018

1 Schéma d'architecture système

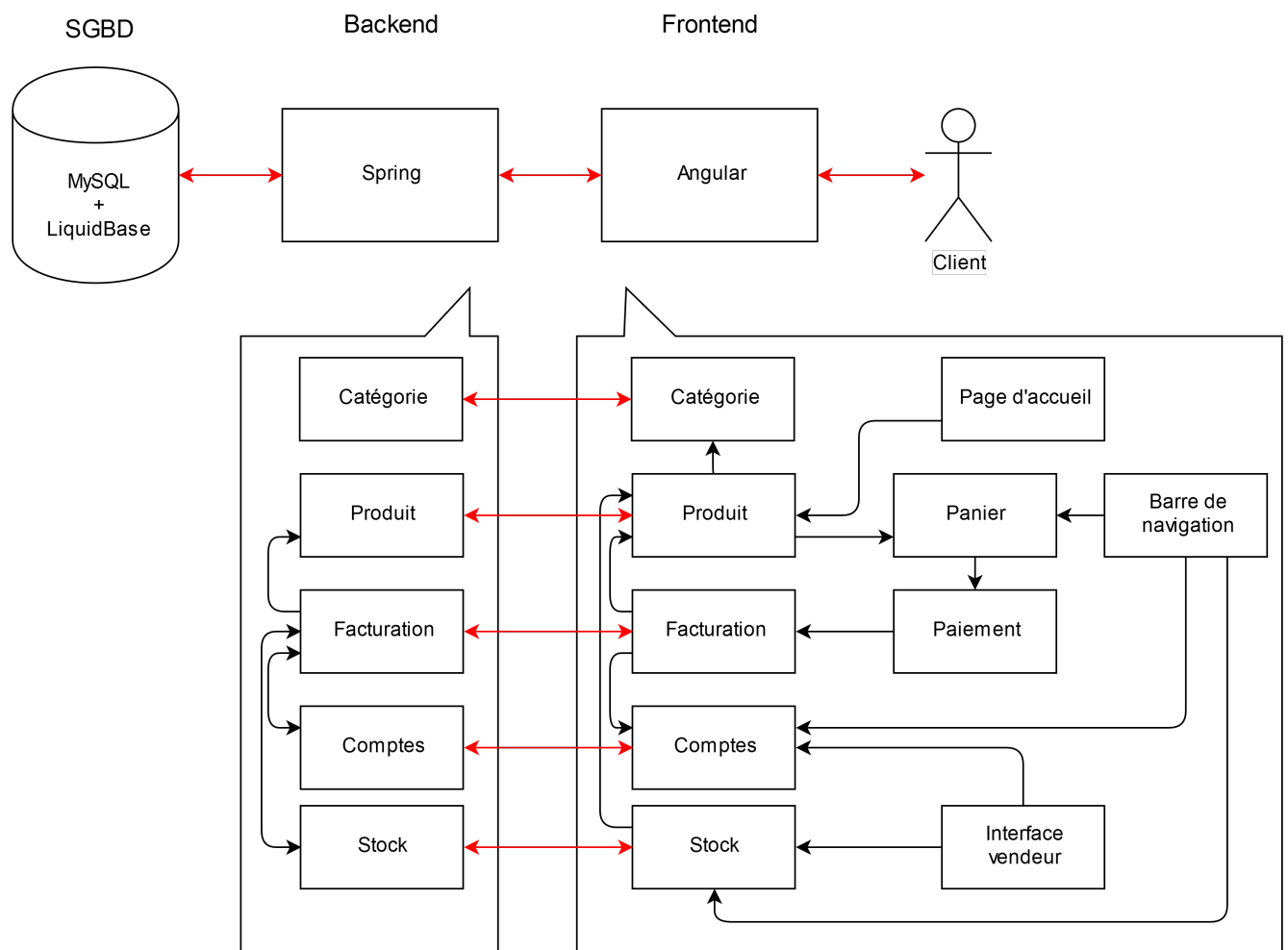


FIGURE 1 – Schéma d'architecture système

2 Schéma de la base de données

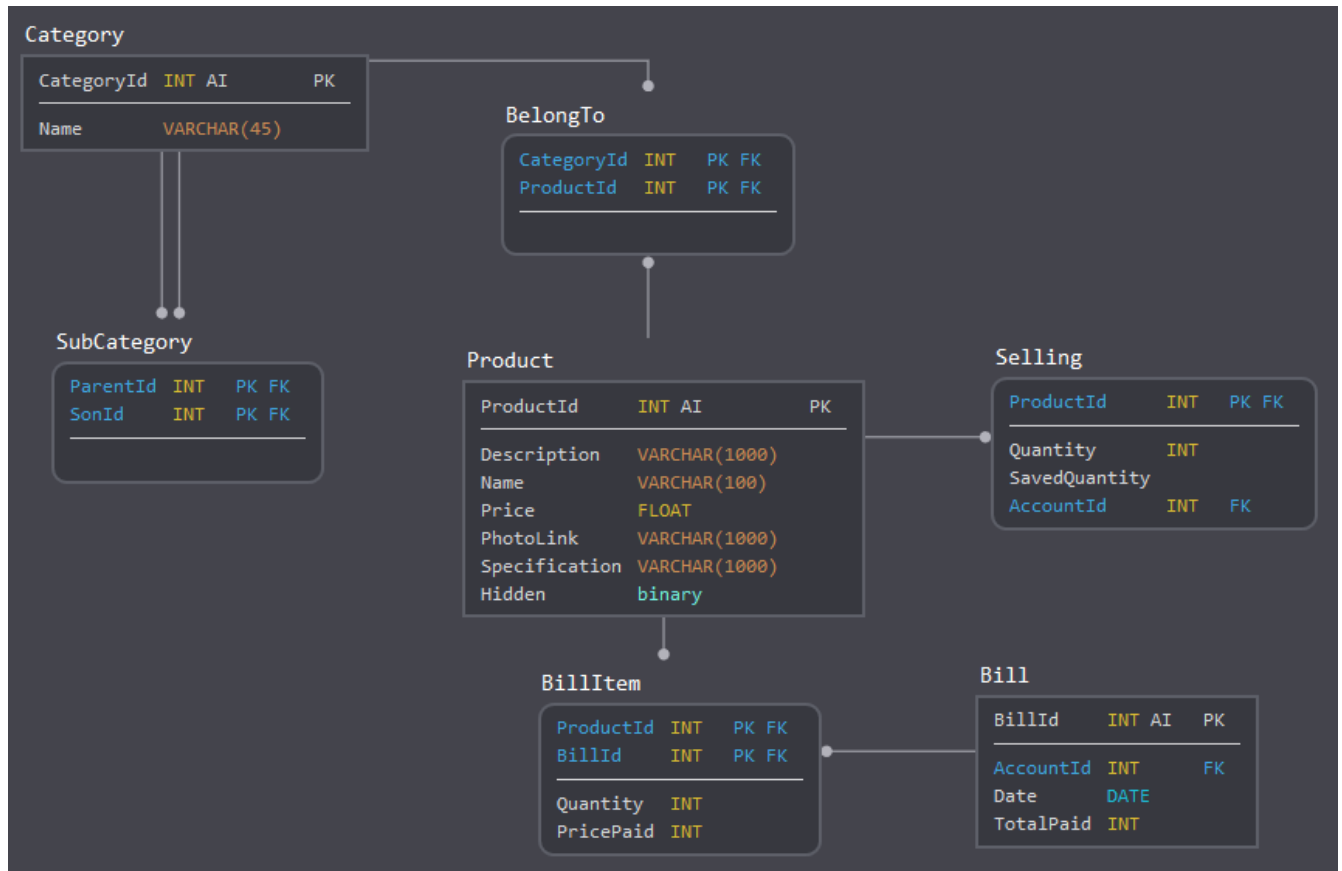


FIGURE 2 – Schéma de la base de données

Note : Cela n'inclut pas le schéma des comptes, qui est celle générée par défaut par Jhipster.

3 Backend

3.1 Produit

Un produit est composé d'un Identifiant, d'un nom, d'une description, d'une spécification, d'un prix et d'un lien vers une image. Un produit peut appartenir à un ensemble de **catégories**.

3.1.1 Accès par l'API REST

- [GET /api/products/\[id\]](#) : renvoie un unique produit dont l'id correspond à celui passé dans le chemin.
- [GET /api/products?\[name\]&\[categorieid\]&\[page\]](#) : renvoie une page de produits de taille 20. Les paramètres après le ? sont tous optionnels et permettent de filtrer la recherche :
 - categorieid permet de chercher les produits appartenant à une catégorie identifiée par son nom.
 - name permet de faire un filtre sur une partie de nom de produit.
 - page permet de sélectionner quelle page de résultat est souhaitée.
- [POST /api/products](#) : crée un produit et l'enregistre dans le SGBD.

3.2 Facturation

Cette partie n'a pas pu être implémentée, les informations suivantes sont donc une prévision de ce qui devrait être fait.

Une facture contient un Identifiant, une référence vers le compte ayant effectué l'achat, la date d'achat et un ensemble d' "objet facturé" (qu'on imagine être composé d'un produit, un prix au moment de l'achat et d'une quantité).

3.2.1 Accès par l'API REST

- [GET /api/factures/](#) : retourne une page des factures d'un compte client. Celui devrait idéalement pouvoir être identifié via son token d'authentification, mais à défaut on pourra imaginer une version plus simple prenant directement un identifiant client en paramètre.
- [POST /api/factures/](#) : crée une facture.
- [GET /api/factures/facture/\[id\]](#) : retourne la facture identifiée par l'id passé dans le chemin.

3.3 Comptes

Nous utilisons ici la base générée par Jhipster, mais nous avons ajouté l'autorité 'vendeur', qu'un utilisateur peut choisir lors de son inscription. Cela n'a pas engendré de modifications sur l'API REST, qui ne sera donc pas détaillée ici.

3.4 Stock

Un stock est identifié par l'identifiant du produit qu'il représente. Il contient également l'identifiant du compte qui vend ce produit, la quantité de produits et la quantité "réservée". Nous adoptons une approche 'optimiste' du stock : lorsqu'un utilisateur ajoute un produit à son panier, celui-ci est 'réservé' pendant quelques minutes, les autres utilisateurs accédant au stock de ce produit voient alors que la quantité disponible (quantité - quantité réservée) a diminuée.

3.4.1 Accès par l'API REST

- [GET /api/stocks](#) : retourne la liste des catégories. On note que contrairement au reste de l'API, on retourne ici la liste de l'intégralité des catégories, car l'utilisateur doit pouvoir choisir n'importe quelle catégorie pour son filtre.
- [GET /api/stocksID](#) : retourne une page des identifiants de produits présent dans un stock.
- [GET /api/stocksUser?\[id\]](#) : retourne une page des stocks du vendeur dont l'identifiant est passé en paramètre.
- [GET /api/stocks/\[id\]](#) : retourne un stock.
- [GET /api/stocks/](#) : retourne la liste des stocks.
- [PUT /api/stocks](#) : met à jour un stock.

3.5 Catégories

Une catégorie contient un Identifiant, un nom, et un ensemble de sous-catégories (on a donc une sorte d'arborescence que l'on comptait exploiter, mais cela n'a pas pu être réalisé).

3.5.1 Accès par l'API REST

[GET /api/categories](#) : retourne la liste des catégories. On note que contrairement au reste de l'API, on retourne ici la liste (et non pas une page) de l'intégralité des catégories, car l'utilisateur doit pouvoir choisir n'importe quelle catégorie pour son filtre.

4 Frontend

Ne sont détaillés ici que les composants ne faisant pas partie de la base générée par Jhipster.

4.1 Product

4.1.1 ProductList

Cette page est composée d'un composant Sidebar et d'une liste de composants SimpleView (un par produit).

4.1.2 DetailView

4.1.3 SimpleView

Ce composant est un affichage simple du produit (image,nom et prix). Il contient également un bouton d'ajout au panier, et un clic sur l'image redirige vers une page chargeant le composant DetailView.

4.2 Categorie

4.2.1 Sidebar

La sidebar est à côté de la liste des produits et permet de les filtrer par catégorie, ainsi que de changer de page de produits.

4.3 Paiement

4.3.1 InfoPaiement

Cette page contient un formulaire de paiement et redirige vers le composant FinPaiement une fois celui-ci rempli et validé.

4.3.2 FinPaiement

Cette page, très simple, informe que la commande a été passée.

4.4 Panier

4.4.1 SmallPanier

Ce composant est situé sur la barre de navigation et redirige sur une page qui charge le composant CartView.

4.4.2 DetailPanier

Ce composant affiche les détails d'un objet du panier et permet d'en changer la quantité.

4.4.3 CartView

Ce composant est un ensemble de DetailPanier, un par objet du panier. il est également possible de vider le panier, ou passer la commande (ce qui redirige alors vers le composant InfoPaiement).

4.5 Stock

4.5.1 SmallStock

Ce composant est situé sur la barre de navigation et redirige sur une page qui charge le composant ListStock.

4.5.2 DetailStock

Ce composant affiche les détails d'un objet Stock et permet de changer ce stock.

4.5.3 ListStock

Ce composant est un ensemble de DetailStock.

4.6 Base Jhipster

Nous ne détaillons ici que les composants que nous avons modifiés.

4.6.1 Home

La page d'accueil charge le composant ProductList.

4.6.2 NavBar

La barre de navigation contient les composants SmallStock et SmallPanier. Elle dispose aussi d'une barre de recherche permettant de charger ProductList avec le filtre de nom.

5 Architecture matérielle

Nous avons déployé notre application sur Google Cloud à l'aide de Google App Engine. Les images sont stockées sur un Amazon S3, le SGBD est stocké sur Google Cloud SQL.

Le site est accessible à l'url <https://ikebay-2.appspot.com/>